Advanced Regression: Linear and generalised linear models I

Garyfallos Konstantinoudis

Feb 21, 2025

Learning Objectives

After this session students should be able to:

- Repeat the basic concepts of linear regression

Main goal of (linear) regression

Regression models are used to investigate association between

- an outcome variable y

- potential explanatory variables (or predictors) $x = (x_1, x_2, ..., x_p)$

The statistical idea is to see if the $x = (x_1, x_2, ..., x_p)$ can give an adequate description of the variability of the outcome y.
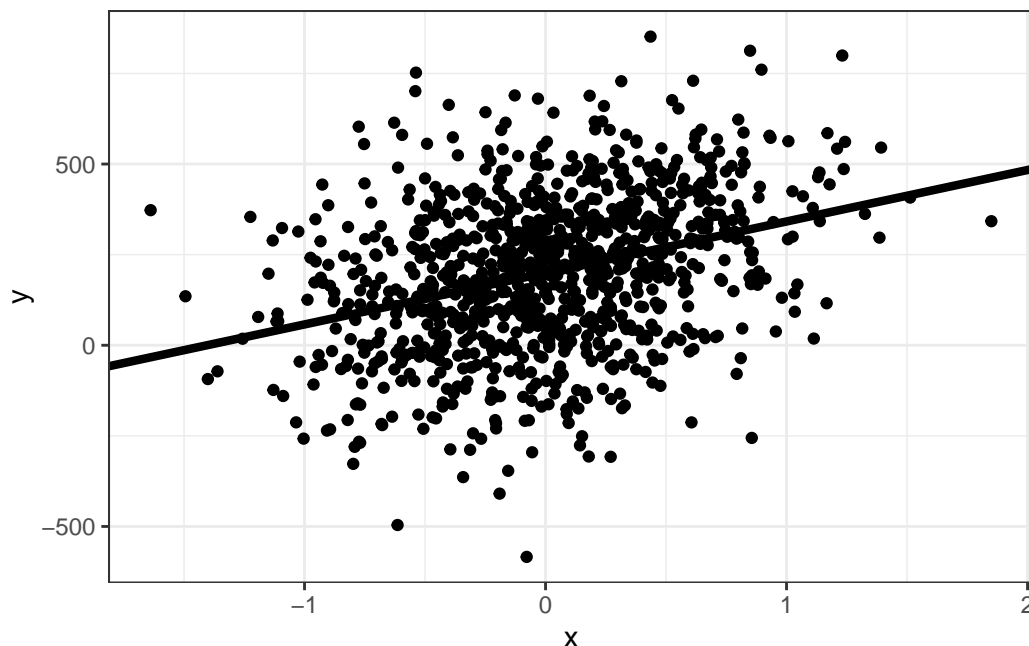
Motivations

1. **Understand** how the predictors affect the outcome.

   - Example: We conduct an observational study focusing on type 2 diabetes as outcome. Our aim is to understand which risk factors are associated with the risk of type 2 diabetes.

2. **Predict** the outcome of new observations, where only the predictors are observed, but not the outcome.

   - Example: We study type 2 diabetes and want to predict disease progression. Our aim is to identify individuals with poor prognosis and improve their treatment.

The linear model

$$y = \alpha + x\beta + \epsilon$$

- y: Outcome, response, dependent variable. Dimension: $n \times 1$

- x: Regressors, exposures, covariates, input, explanatory, or independent variables. Dimension: $n \times p$

- ϵ: Residuals, error. Dimension: $n \times 1$

2

- α: Intercept. Dimension: 1×1

- β: Regression coefficients. Dimension: $p \times 1$

Parameters to estimate:

- α: Intercept, Baseline level, the expected mean value of y when all $x = 0$

- $\beta = (\beta_1, ..., \beta_p)$: vector of regression coefficients.

- β_j: regression coeffcients of variable x_j. The expected change in y for a one-unit change in x_j when the other covariates are held constant.

Observed data:

- y: Outcome or response.

- x: Regressors, exposures, covariates, input, explanatory or independent variables

    - $i = 1, ..., n$ samples.

    - $j =, ..., p$ variables.

Estimates: Ordinary least squares (OLS)

$$\hat{\beta}_{OLS} = \underbrace{(x^t x)^{(-1)}}_{p \times p} \underbrace{x^t}_{p \times n} \underbrace{y}_{n \times 1}$$

- Inversion of $\underbrace{(x^t x)^{(-1)}}_{p \times p}$ requires $x^t x$ to be of full rank (Lecture 2b).

Alternative representation:

- $\hat{\beta} = \frac{c\hat{o}v(x,y)}{c\hat{o}v(x)}$, where the sample covariance is defined as:

    - $c\hat{o}v(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$
    - $c\hat{o}v(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})$

Example: Diabetes data

- y : quantitative measure of disease progression one year after baseline (vector)
- x: predictor matrix

- clinical parameters: age, sex, bmi

- map: blood pressure

- tc: total cholesterol

- ldl: low-density lipoprotein

- hdl: high-density lipoprotein

- tch: total cholesterol over hdl

- ltg: triglycerides

- glu: glucose

- $n = 442$: sample size

The lm() command in R

```
lm(y ~ age + sex + glu + map + ltg, data = x)
```

Formula:

```
y ~ x1 + x2 + x3
```

- left of $\sim$: outcome
- right of $\sim$: predictors

It is also possible to enter a full matrix x, transform by as.matrix(), as multivariable set of predictors:

```
y ~ x
```

An intercept is always included, to turn off add -1

Interpreting the summary.lm() command

```r
library(lars)
```

```
Loaded lars 1.3
```

```r
library(dplyr)

data(diabetes)
x <- as.data.frame.matrix(diabetes$x)
y <- diabetes$y

lm(y ~ age + sex + glu + map + ltg, data = x) %>% summary.lm()
```

```
Call:
lm(formula = y ~ age + sex + glu + map + ltg, data = x)

Residuals:
     Min      1Q  Median      3Q     Max
-165.128 -43.025  -5.232  42.446 182.050

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  152.133      2.871  52.990  < 2e-16 ***
age          -54.227     65.854  -0.823  0.41071
sex         -166.066     62.903  -2.640  0.00859 **
glu          175.377     71.671   2.447  0.01480 *
map          426.532     70.342   6.064 2.89e-09 ***
ltg          706.395     70.929   9.959  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 60.36 on 436 degrees of freedom
Multiple R-squared:  0.3939,    Adjusted R-squared:  0.387
F-statistic: 56.68 on 5 and 436 DF,  p-value: < 2.2e-16
```

Difference between univariable and multivariable regression

```r
lm(y ~ glu, data = x) %>% summary.lm()
```

```
Call:
lm(formula = y ~ glu, data = x)

Residuals:
     Min       1Q   Median       3Q      Max
-153.069  -57.716   -5.466   54.656  186.839

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  152.133      3.392  44.851   <2e-16 ***
glu          619.223     71.312   8.683   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 71.31 on 440 degrees of freedom
Multiple R-squared:  0.1463,    Adjusted R-squared:  0.1444
F-statistic:  75.4 on 1 and 440 DF,  p-value: < 2.2e-16
```

Reduction of the regression coeffient from 619 to 175 after conditioning on other covariates $\rightarrow$ attenuation of the effect

Further estimates

- Weighted least squares

$$\hat{\beta}_{WLS} = \underbrace{(x^t w x)}_{p \times p} \underset{p \times n}{x^t} \underset{n \times n}{w} \underset{n \times 1}{y}$$

  where w is a $n \times n$ diagonal weight matrix

- Maximum likelihood

- Bayesian linear regression (Module: Bayesian Statistics)

Fitted values and residuals

- Fitted values

$$\hat{y} = x\hat{\beta} = \underbrace{x(x^t x)^{-1} x^t}_{h} y$$

- Hat matrix h

- Residuals are the difference between the fitted values (predicted by the model) and the actual observed outcome: $r_i = \hat{y}_i - y_i$

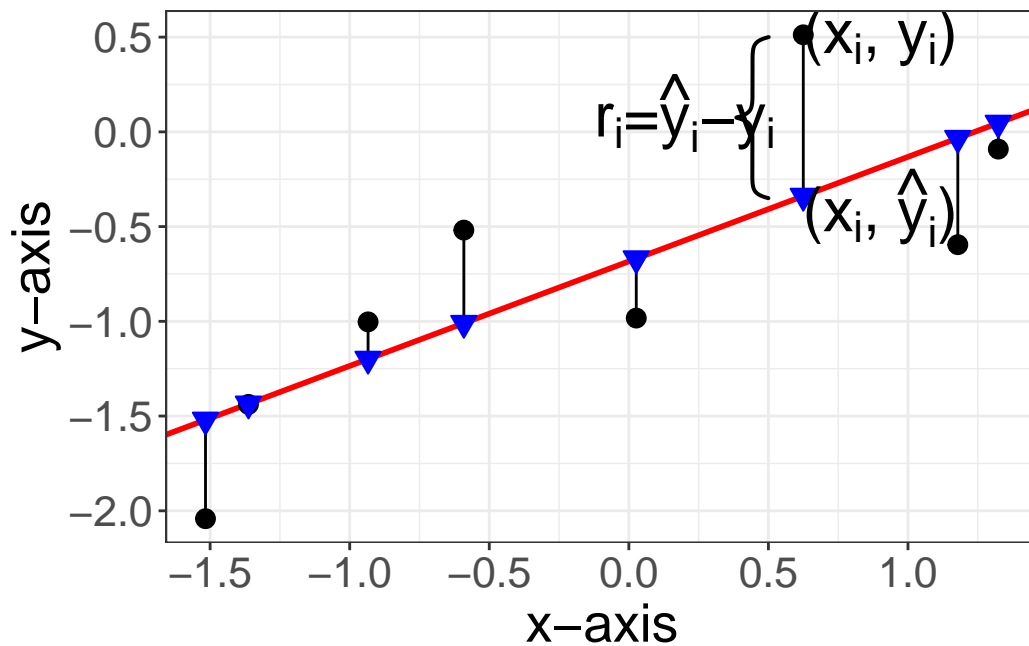- The residuals are a vector $r = (r_1, ..., r_n)$ of length n.

> 💡 Tip
>
> Residuals are an important quantity for model diagnostics.

Fitted values and residuals

```
Warning in is.na(x): is.na() applied to non-(list or vector) of type
'expression'
Warning in is.na(x): is.na() applied to non-(list or vector) of type
'expression'
Warning in is.na(x): is.na() applied to non-(list or vector) of type
'expression'
```

lm(): Fitted values and residuals

- First fit a linear model and save it in the object lm0

```
lm0 <- lm(y ~ glu, data = x)
```

- The linear model object lm0 contains
    - Regression coefficients

```
lm0$coefficients
```

```
(Intercept)          glu
   152.1335     619.2228
```

    - Fitted values

```
lm0$fitted.values
```

    - Residuals

```
lm0$residuals
```

8

Assumptions

1. Linearity: There is a linear relationship between x and y.
2. Weak exogeneity: The predictors x are viewed as fixed variables; there is no measurement error on x.
3. Constant variance (homoscedasticity): All residuals have the same variance.
4. No perfect multicollinearity: No predictor can be expressed as a linear combination of the other predictors (Lecture 2b).
5. Independent errors: The residuals are uncorrelated (e.g. in time-series the error of time point t will depend on the error of time point $t-1$) and independent of x.

Further assumptions

- Normal-distributed errors:

  The residuals are normal-distributed.

  Note: This is not required for the OLS estimate, but for the Maximum Likelihood estimation.

- Outlier: observation point that is distant from other observations.

  It is recommended to check the data for outliers, which can arise because of many reasons:

  – Measurement error (remove)

  – Errors in the pre-processing steps (fix or remove)

  – "True" biological outliers (follow-up)
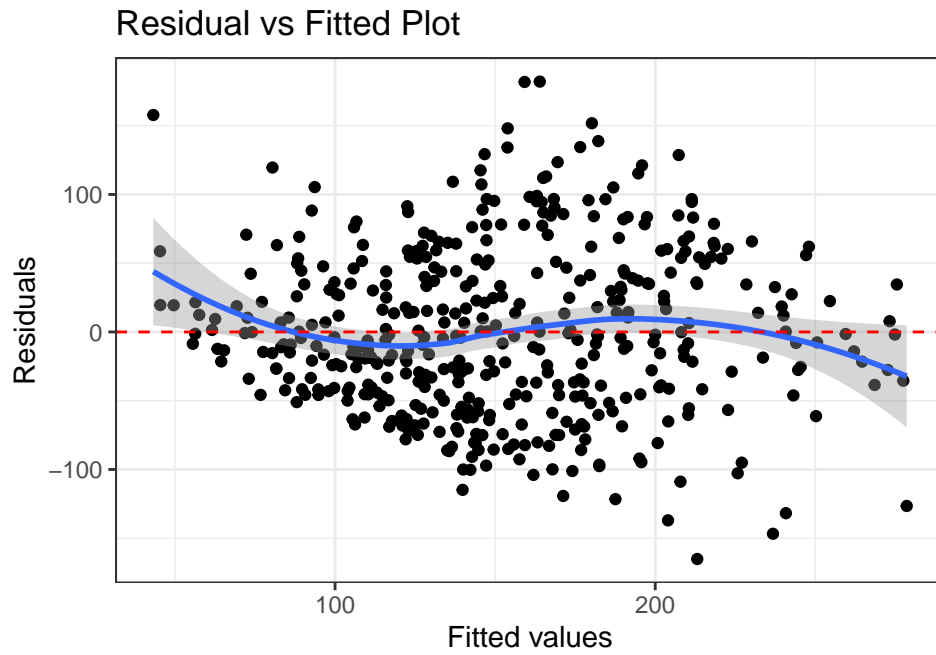
- Influential variants: Cook's distance

Diagnostic plots: Linear relationship
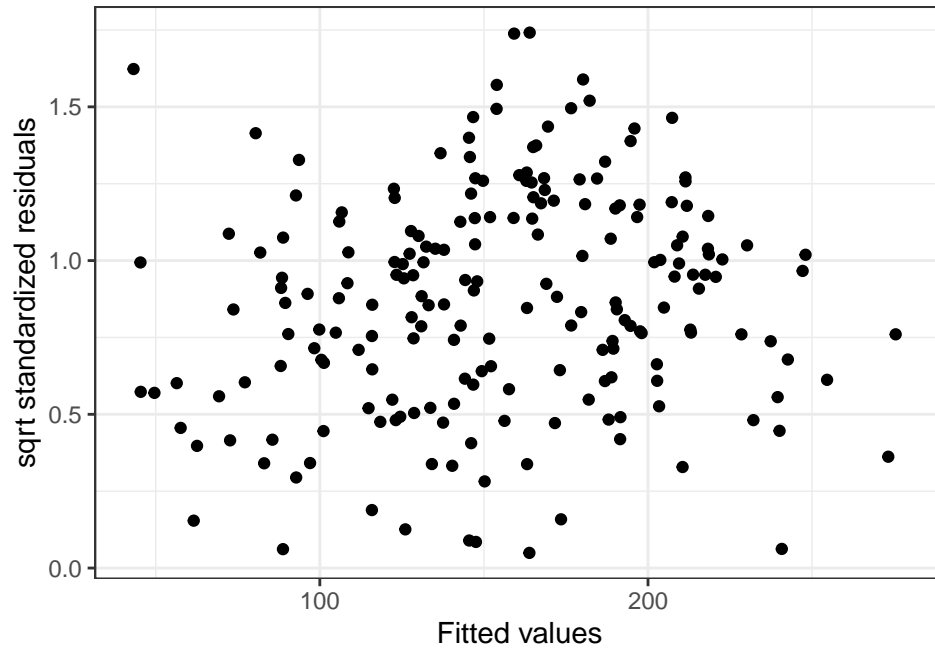
- Scatterplot of y against x
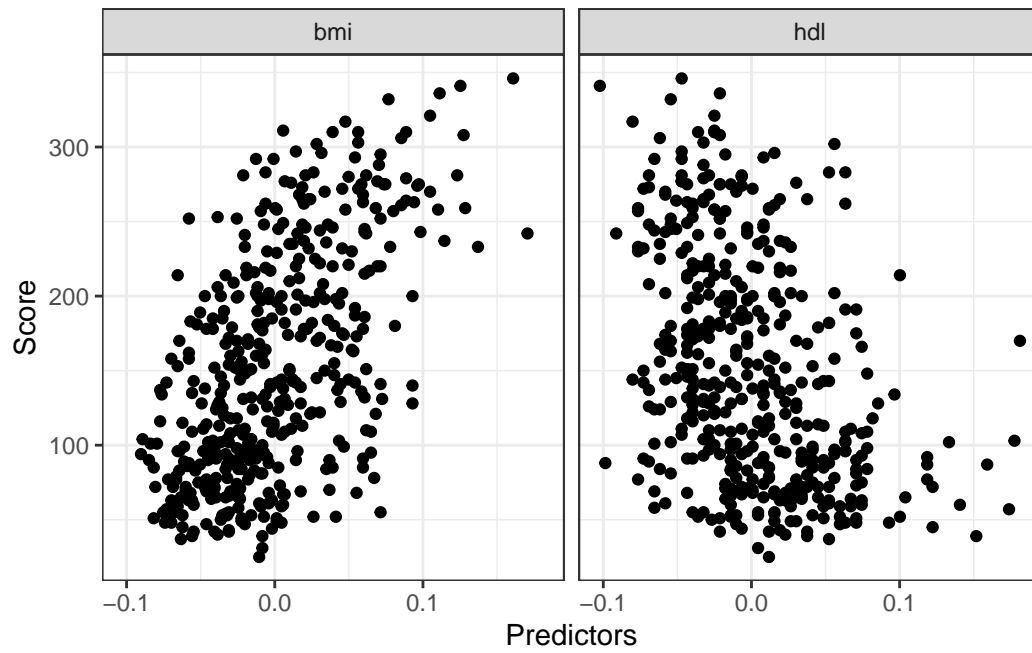
Diagnostic plots: Linear relationship

- Scatterplot of residuals (y-axis) against fitted values (x-axis)
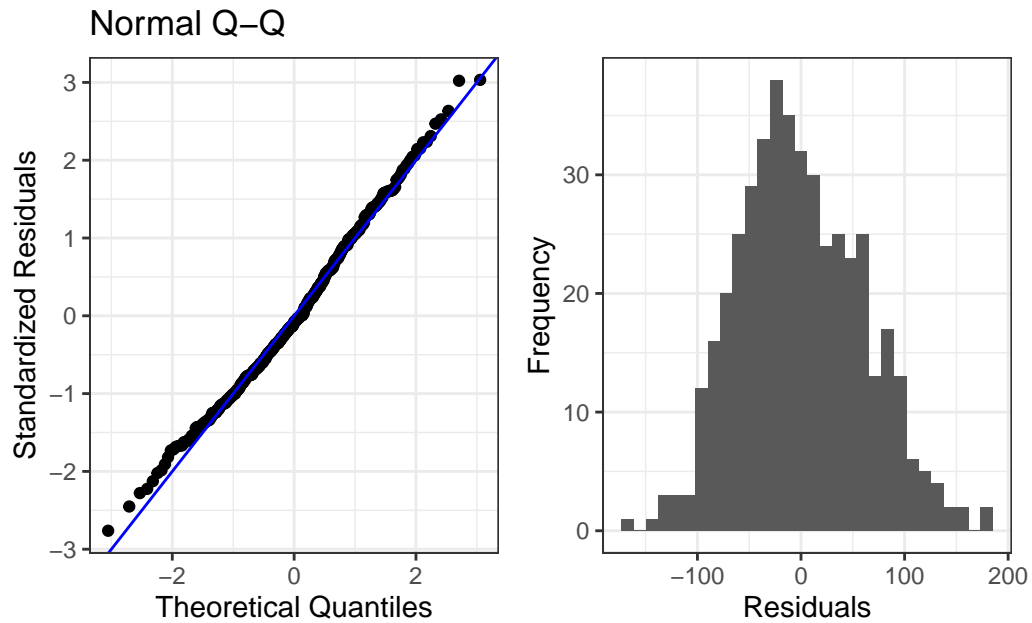
Residual vs Fitted Plot

Diagnostic plots: Homoscedasticity

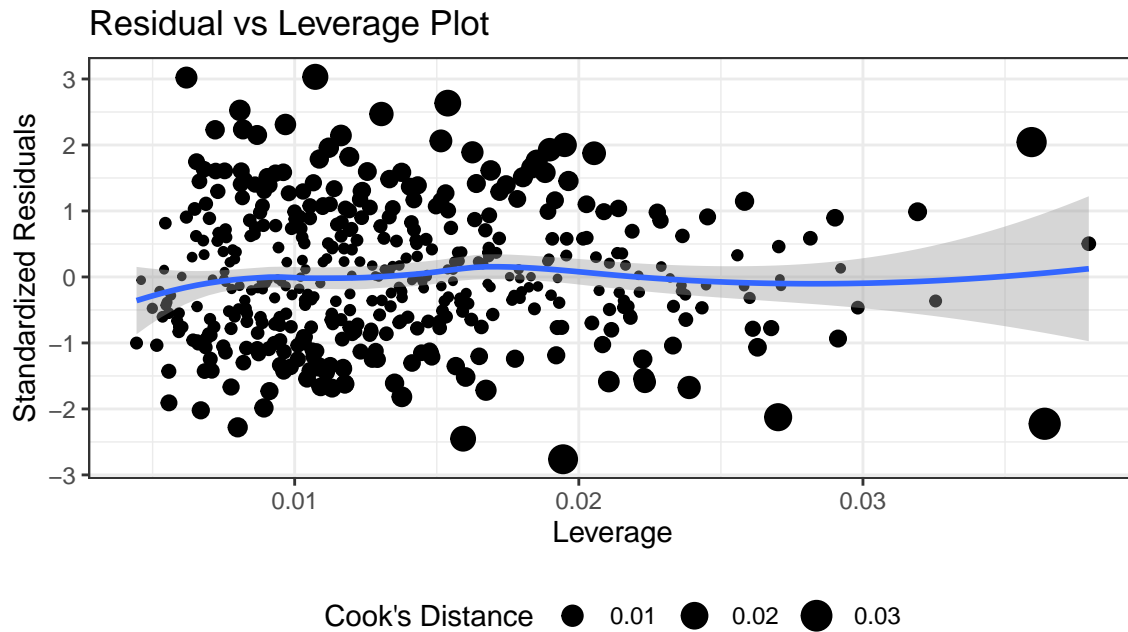- Scatterplot of standardised residuals (y-axis) against fitted values (x-axis)

Diagnostic plots: Normal-distribution of residuals

- Q-Q plots of observed residuals (y-axis) against theoretical values under the Normal distribution (x-axis)

Diagnostic plots: Outliers

- Scatterplot of standardised residuals against Cook's distance.

- Cook's distance measures the effect of deleting a given observation (sum of all the changes in the regression model when observation i is removed).

Residual vs Leverage Plot

lm(): Diagnostics

- Linear relationship and outliers (Scatterplot of y against x)

```
plot(x, y)
abline(mod, col = "red")
```

- Linear relationship and outliers (Residuals against fitted values)

```
plot(mod, which = 1)
```

- Homoscedasticity (Standardised residuals against fitted values)

```
plot(mod, which = 3)
```

- Normal-distribution of residuals (Q-Q plots of observed residuals against theoretical values under the Normal distribution)

```
plot(mod, which = 2)
```

- Influential variants: (Standardised residuals against Cook's distance)

```
plot(mod, which = 5)
```

Prediction using linear models

Assume we have a database with n type 2 diabetes cases, where we have measured the following data:

- y: quantitative measure of disease progression one year after baseline (vector)

- x: predictor matrix including clinical data (age, sex, bmi), blood pressure and triglycerides

- This is our training data y_train and x_train.

For a new case we only have the predictor matrix x_{new}, but not y_{new}.

Goal: For each new type 2 diabetes case we want to predict y_{new}, his/her progression one year later.

lm(): Predictions

- Use the linear model to learn a prediction rule from the training data, where both x and y are observed on the same individuals.

```
lm_train <-
    lm(formula = y_train ~ age + sex + bmi + map + ltg, data = x_train)
```

- Predict the outcome based on the prediction rule and the predictors of the new data.

```
predict.lm(lm_train, x_new)
```

Take away: Linear models

- Motivation why to use linear models (To understand and to predict)

- Model fit using ordinary least squares

- Interpretation of the regression coefficients

- Residuals and fitted values

- Model diagnostics

- Using the linear model to predict

Questions?